

VISUALISING LEARNING DESIGN IN LAMS: A HISTORICAL VIEW

by **James Dalziel**

Macquarie University

james.dalziel @ mq.edu.au

Abstract

The Learning Activity Management System (LAMS) provides a web-based environment for the creation, sharing, running and monitoring of Learning Designs. A central feature of LAMS is the visual authoring environment, where educators use a drag-and-drop environment to create sequences of learning activities. The visualisation is based on boxes representing discrete activity tools (forum, chat, quiz, content, etc.) which are connected together using arrowed lines to indicate the flow of tasks. This visual approach to authoring of Learning Design has both strengths and weaknesses: in terms of strengths, it has provided a common visual language among LAMS users for rapid adoption and sharing of instructional strategies, and a useful framework for simple linear pedagogical approaches; in terms of weaknesses, the visual simplification necessarily limits the amount of information that can be conveyed about a complex instructional design, especially those designs not easily adapted to a linear format (eg, spiral pedagogies). This paper describes the assumptions behind the LAMS visual authoring environment at the levels of both educational theory and software design, together with a review of implementation experiences among educators, including experiences from the LAMS Community. The paper concludes with reflection on future directions for visualisation of Learning Design, particularly in the area of annotation and time-based visualisation.

1. Overview

The Learning Activity Management System (LAMS)[1] is a web-based Learning Design system first implemented with teachers and students in 2003 (Dalziel, 2003a) that continues to be developed to the present day. LAMS is used by thousands of educators in over 80 countries, is translated in 30 languages, and as at April had an online community of practice (the LAMS Community) with 5,753 members and 595 community-shared designs which had been downloaded/previewed 17,375 313 times (Dalziel, 2010).

LAMS provides an integrated Learning Design system, incorporating an authoring environment, a run-time implementation environment for students (including a suite of activity tools – 10 in the initial V1 release, 24 in the V2.3.4 release) and a monitoring environment for teachers to track real-time student progress. The software has been through two major generations of development (Version 1: 2003-2006; Version 2 – a complete rewrite of the system: 2007-present) with the current version at the time of writing being V2.3.4.

A Learning Design authored in LAMS (typically called a “sequence”) can be exported as a file (XML only in V1; XML and resources in a zipped file in V2) which allows it to be shared with other educators either directly or through a website such as the LAMS Community. Due to problems encountered during implementation, the typical export format for LAMS sequences is similar to, but not the same as, the IMS Learning Design specification (Dalziel, 2006), although an IMS Learning Design Level A export is available as an option.

2. Background – software design

Following the work of Rob Koper and colleagues on Educational Modelling Language (EML) (Koper, 2001), the IMS Global Learning Consortium adopted EML as the foundation for the IMS Learning Design (IMS LD) specification (IMS, 2003). Even while the specification was being finalised, there was active development of software systems inspired by the ideas of EML and IMS LD.

At the third meeting of the Valkenburg Learning Design implementation group in Paris, October 2002 (see Campbell, 2002), I indicated that the focus of our work would be on building a system that incorporated functioning activity tools within its run-time framework. At the time there was a general sense that this was not a good strategy for system development, as it was expected that Learning Management System (LMS) developers would configure their existing activity tools to work in Learning Design systems. I also indicated our intention to build a visual authoring environment to complement the run-time environment (Dalziel, 2002). The combination of authoring and run-time environments within a single system was also a departure from recommended approaches (Olivier, 2004).

These early decisions proved to have an impact on the development of LAMS, particularly the nature of its visual authoring environment. Rather than take the approach of defining an XML specification first (with all the implied richness of anticipated system functionality that this may entail), LAMS development started with the more narrow scope of actual capabilities of a run-time environment (incorporating activity tool features), and then “worked backwards” to a visual authoring environment for creating sequences of activities that could be instantiated within the existing run-time environment.

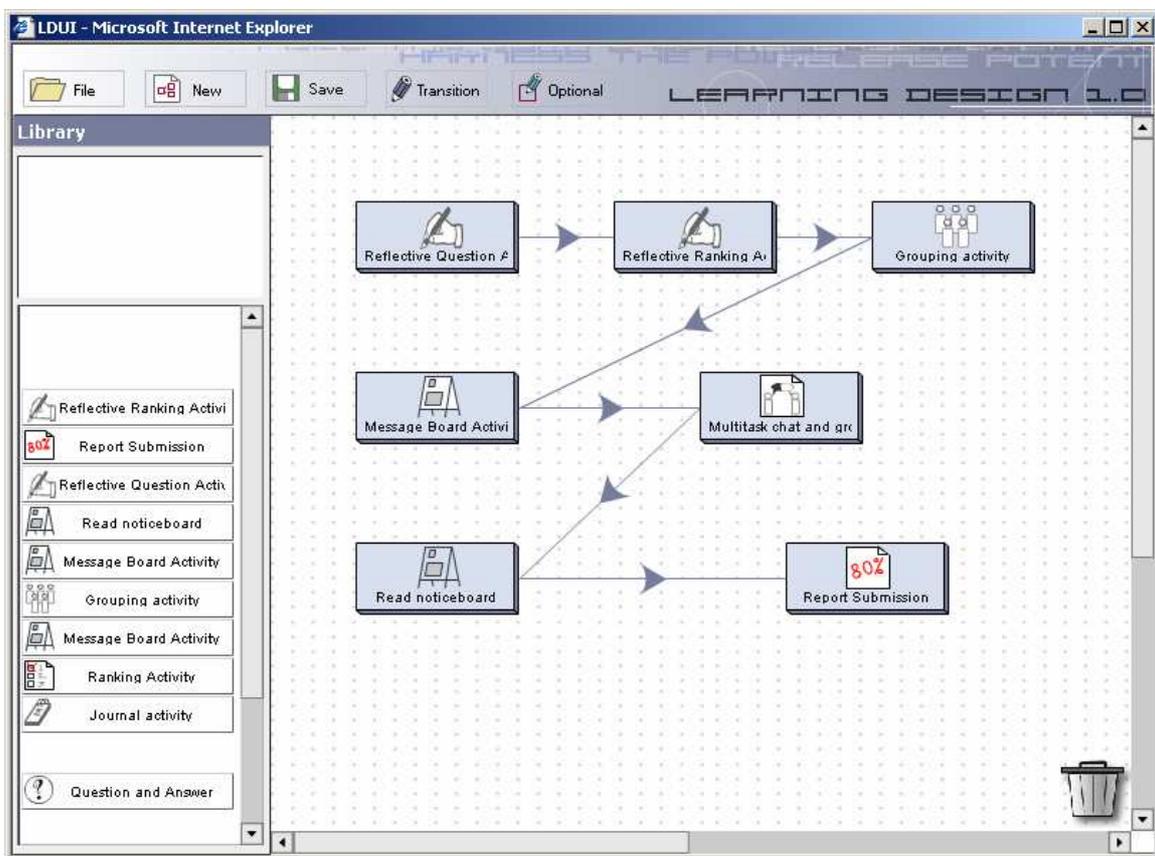


Figure 1. Screenshot of LAMS Authoring environment from first public demonstration at Valkenburg Group meeting, February 2003.

This approach had both significant negative and positive effects on the early visual authoring environment of LAMS (i.e., Version 1 of LAMS). A key negative effect was to greatly constrain the range of possibilities of EML/IMS Learning Design to those learning designs with linear structures based on a suite of typical LMS-like tools (see Figure 1 for the first public example of the LAMS Authoring environment demonstrated

at a Valkenburg Group meeting in Vancouver in February 2003 – from Dalziel, 2003b). The authoring environment did not provide structures for learning designs with multiple pathways (eg, automated branching by the system based on prior behaviour, or student-selected multiple pathways) or those with non-linear approaches, such as spiral-style pedagogical approaches (eg, creative writing activities which involve iterative development and refinement of a text).

One small exception to the linear requirement was a feature for “Optional Activities”, where students could choose from among several different activities – but this was only a choice from among single activities at a specific point in a sequence of otherwise linear activities (see example in Figure 2). To support management of linear flows of activities, LAMS provided a “Stop Point” feature which allowed the Monitor to control the timing of release of subsequent activities to students.

A second limitation arose in the area of roles. LAMS provided a teacher and student role, with relevant functional differences in activity tool behaviour – eg, if answers to a Q&A were designated as anonymous during LAMS authoring, then at runtime, student answers were presented anonymously to students, but the teacher had a separate interface which showed the name of each student with their answer. However, LAMS did not provide automation for more sophisticated role structures, as is anticipated in IMS Learning Design (IMS, 2003). One exception to this in LAMS was the “Chat and Scribe” tool which included functionality for two student roles: a general discussion role for all students, and a special “scribe” role for one student who could edit answers to questions based on group discussion (for an illustration of this tool, see http://saturn.melcoe.mq.edu.au/lams2/docs/winks/activities/c1_chat&scribe.htm).

For completeness, it should also be noted that role descriptions could be included as textual instructions in the absence of system functionality (eg, a group of students in a forum could designate one student as the “reporter” to provide input back to the class even if the system was unaware of this role at a functional level).

Taken together, the negative impact of these limitations was to constrain LAMS authoring (in Version 1) to simple linear pedagogies with only basic roles. Nonetheless, it was possible to build some interesting pedagogical approaches within these constraints, as a 2003 example in Figure 2 of the IMS LD specification Versailles role play illustrates (Dalziel, 2003c) (although it should be noted that when implemented in

LAMS, this sequence had two weaknesses compared to the IMS LD narrative description – it did not forbid students entering the negotiation areas of other groups, and it only provided random allocation of students to role groups, not teacher selection).

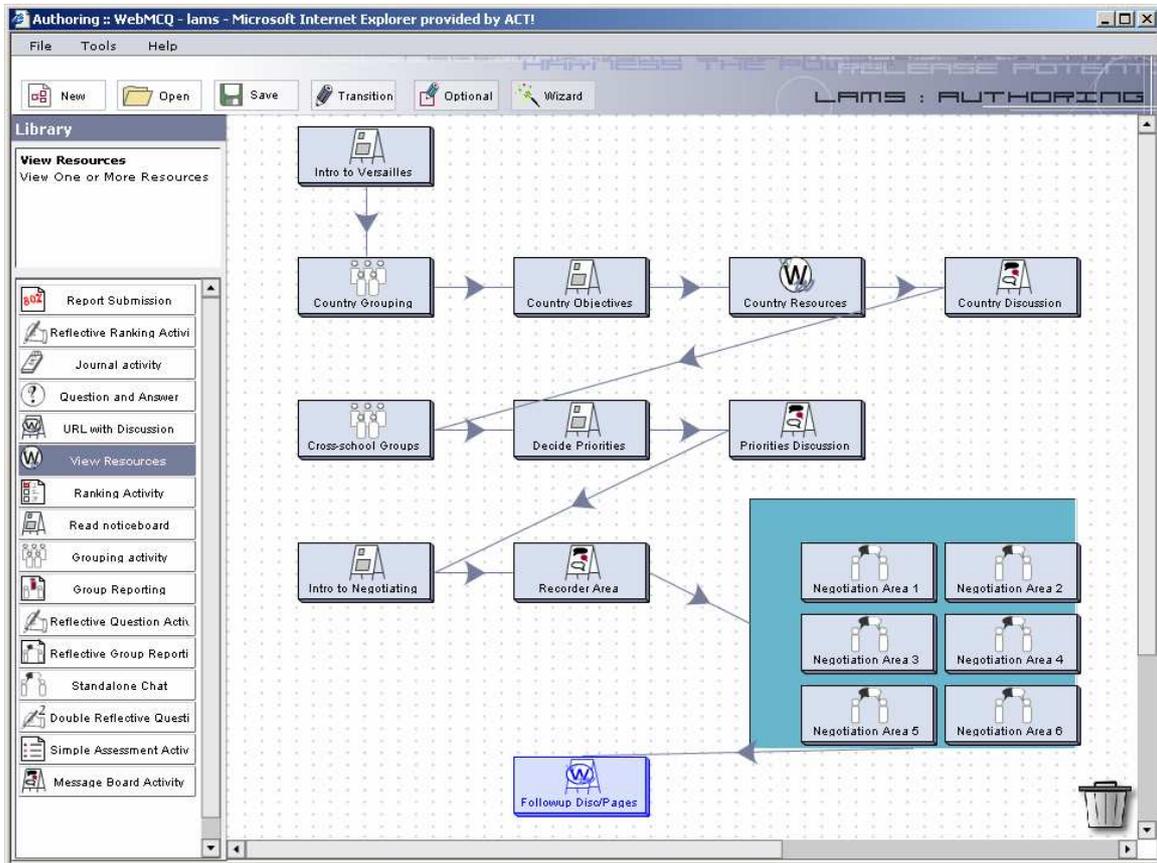


Figure 2. Versailles role play from IMS LD specification modelled in LAMS V1, presented at CETIS Pedagogy meeting August 2003.

A positive effect of the decision to “work backwards” from run-time functionality to the authoring environment was that an authored design could be run with students, and authors could use a “Preview” mode to immediately see the student experience of the authored design, and change or extend their design based on the preview. Initially authors could only see a preview of their design by implementing a class session (with a dummy class) – a process that took eight steps – but in early 2005, a one-click “Preview” button was added to the LAMS Authoring environment which allowed for rapid iterative testing and development (Dalziel, 2005).

It is worth noting that the software development challenges of incorporating a “Preview”-style function into a Learning Design authoring system are considerable, as this feature requires an end-to-end infrastructure for instantiation an authored learning design (including activity tools and a sequencing engine). The history of development of other Learning Design systems illustrates the challenges of this requirement; and the ability of LAMS to provide this feature was largely due to the narrowed scope noted above.

Another factor affecting the LAMS visual authoring environment was a decision that no teacher would need to work with XML or UML within LAMS (unlike most other systems, Olivier, 2004). One of the goals of LAMS was for the system to be used by a typical teacher with little training, so an anticipation of prior understanding of XML and UML was unreasonable for this target audience.

A related but more nuanced issue was that no teacher would be expected to have prior understanding of the IMS Learning Design specification in order to use LAMS. While a sequence built in LAMS could be mapped back to the core concepts of IMS Learning Design (method, roles, environments, etc), it was felt that the specification was unduly complicated and recursive to allow for rapid understanding by a typical teacher. Hence there was a need to create an alternative representational system for the concept of Learning Design as instantiated within LAMS without direct reference to the concepts denoted in IMS LD (as illustrated by systems such as ReLoad/ReCourse, Olivier, 2004).

The solution, in the case of LAMS, was to collapse the implied functionality of IMS LD around discrete activity tools, and the sequencing of these tools. Hence the “method” component in IMS LD is reduced to a visual representation of sequencing (in LAMS, this is achieved by drawing lines between activity tool boxes, with lines containing arrows representing the direction of flow over time); while the “environments” component is reduced to the available suite of specific tools, each with their own “tool-relevant” functionalities (ie, while all tools could include general text instructions/information for students, a quiz tool would contain quiz configuration features, while a forum tools would contain thread configuration features, etc). The roles component, as previously noted, was simplified to a general “student” and “teacher” role, with relevant functionality embedded in each activity tool.

Taken together, these simplifications allowed a teacher to use familiar LMS-style tools like chat, quiz, forum, etc, within an authoring environment that created linear flows of tools through the convention of drawing lines between tools to indicate progression over time (represented by arrows embedded within connecting lines).

The resulting visualisation provided information at two levels of granularity – the overview level showing the type of tools selected and their place within the flow of tasks (i.e., the sequence), and then for each individual tool (accessed by double-clicking on the tool icon), detailed information about the content/instructions for the tool, together with relevant functional settings for the chosen tool.

It is interesting to reflect in 2010 (in an era of mashups and workflow systems) just how different this approach was to other approaches to e-learning in 2003. The success of the web illustrated the potential of non-linear display of information where everything is a connected web that the user can navigate at will – and the implicit mental model of a non-linear web of interconnected content was a dominant metaphor in educational technology of this time. Even e-learning content that was presented sequentially to students (such as SCORM courseware) was rarely represented as a linear flow to authors (with Authorware being one notable exception), and the concept of *sequences of collaborative* activities was quite rare.

Hence the drag-and-drop nature of the LAMS Authoring environment, and its implicit visual language, were greatly influenced by underlying technical choices and constraints at both theoretical and practical levels.

3. Background – educational theory

Apart from technical goals and constraints, the visual nature of the LAMS Authoring environment was also driven by certain educational assumptions. At its most basic level, the system was an attempt to implement the concept of Learning Design (defined broadly) – in particular the idea of a system that could represent many different instructional strategies (eg, problem-based learning, role plays, case-based reasoning, web quests, etc) within a single authoring environment.

While the simple linear framework available was unable to capture certain approaches (eg, the spiral pedagogy of the creative writing example given above), a range of different approaches could be represented with some “creativity” in the choice

of tasks and the instructions given to students. For example, use of research tasks (eg, using a Search Engine within the Share Resources tool) combined with an open-ended discussion forum could lead to rich constructivist-style learning experiences despite the fact that the tools were presented in a linear sequence (it is important to note that students could revisit past activities and continue work on them, so a sequence that appeared to be linear in its authoring representation might in fact involve stages of iterative student interaction – see Figure 2 of Takayama *et al.*, 2007).

While the Preview feature was a key technical challenge, its educational goal was broader than simply a “check” on how a finished sequence would appear to students. Instead, educators who used the LAMS Authoring environment frequently reported an experience that seeing a learning design “in action” contributed to an iterative process of design, with some educators reporting a sensation of having implicit decisions in their design process made more conscious as a result of seeing the instantiated outcome (Masterman & Lee, 2005). This iterative use of LAMS Authoring and Preview seems to be the typical use of LAMS – it seems relatively rare that an author designs an entire sequence without making changes as a result of previewing the student experience.

Another educational goal of the LAMS Authoring environment was to create a common means of communication about learning designs among educators (at least for those familiar with the system). During workshops with LAMS authors it became clear that the LAMS environment had provided a common visual language among educators for talking about sequences of learning activities. Even when educators were debating an entirely face to face teaching method, they could use the LAMS representations as a common “shorthand” to make different theoretical point about the best ways to structure sequences of student learning activities.

A less successful dimension of LAMS visualisation was the capacity for denoting the purpose of activities at the overview (sequence) level. Although the text describing each activity tool could be edited (so that the “Noticeboard” tool could be renamed to “Welcome” for an introductory page at the start of a sequence – see Figure 2 for a detailed example), most LAMS sequences shared by authors retain their default tool descriptions (ie, “Noticeboard”, “Forum”, “Chat”, etc), which fails to convey their educational intent. In addition, there was no annotation capability outside of activity

tools to describe the purpose of whole sections of a design (whereas by comparison this is a strong feature of the Compendium Learning Design authoring tool [14]). As a result, when an educator receives a LAMS sequence from another educator, s/he often has to infer the purpose of certain activities from their context and the instructions for students presented inside each tool – this is typically achieved by previewing the sequence.

4. Subsequent development

The greatest limitation of LAMS V1 was the lack of support for multiple pathways, either in the form of system-automated branching or student-selected alternative pathways. The introduction of these features required a complete rewrite of the LAMS software, leading to LAMS V2 (which provided the underlying architecture for these features) and the V2.1 release which provided features for four types of multiple pathways:

- “Branching” (automated)
 - Teacher allocates students to branches at run-time (via Monitor)
 - Group-based (existing student groups are mapped to branches – group membership can be randomly allocated, teacher selected or student selected)
 - Input-based (inputs from prior tools, such as quiz scores, are mapped to branches)
- “Optional Sequences” (student chosen)
 - Students choose to complete one or more multiple pathways

The visualisation for Branching in LAMS V2 treats a set of “branched activities” as a discrete set of activities within an overall sequence, where a summarised branching icon (large box) is created for the branched activities at the sequence level (see Figure 3 for an example), and then by double clicking on this icon, a detailed view of the branching is shown (see Figure 4). While LAMS V2 could support “branches within branches”, it was not well suited to visualising or running more than two layers of branching due to the decision to treat each set of branched activities as a discrete unit. As with other features in LAMS V1, this decision had a significant negative and positive effect.

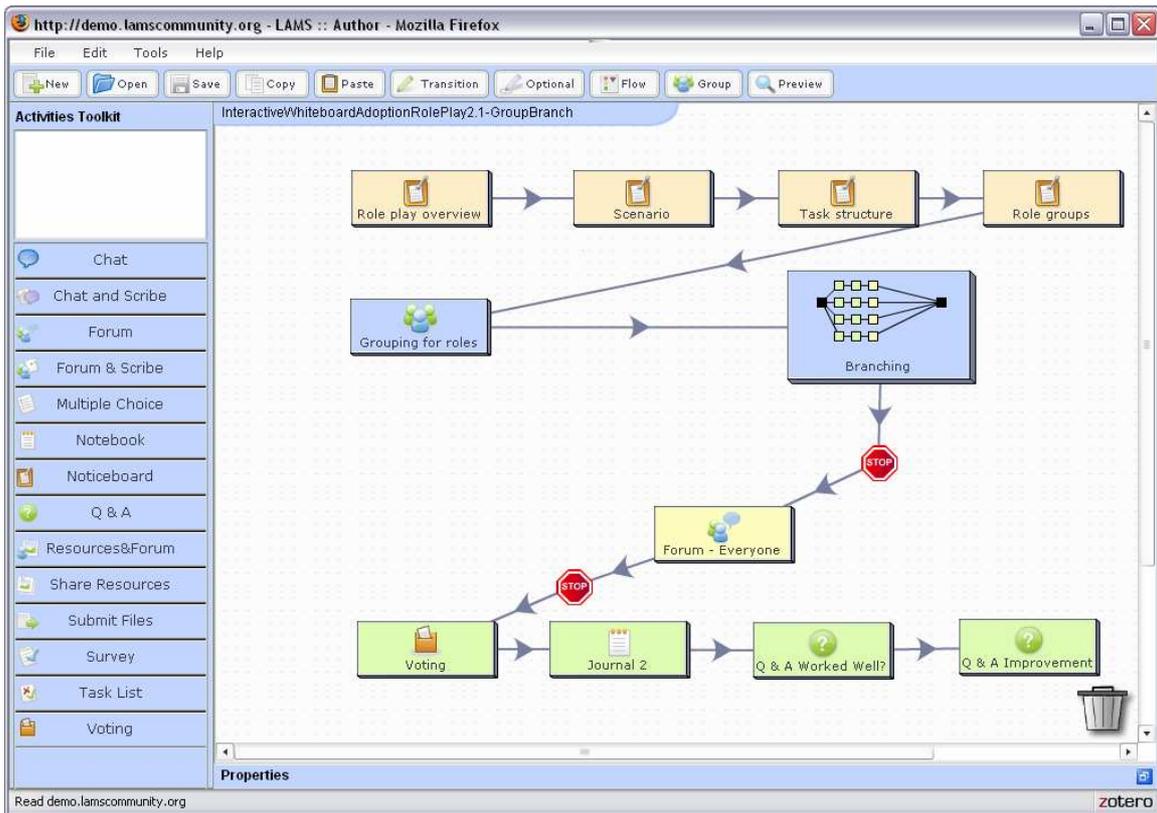


Figure 3. Screenshot of LAMS V2 Authoring environment showing a “role play” sequence, including summary branching visualisation.

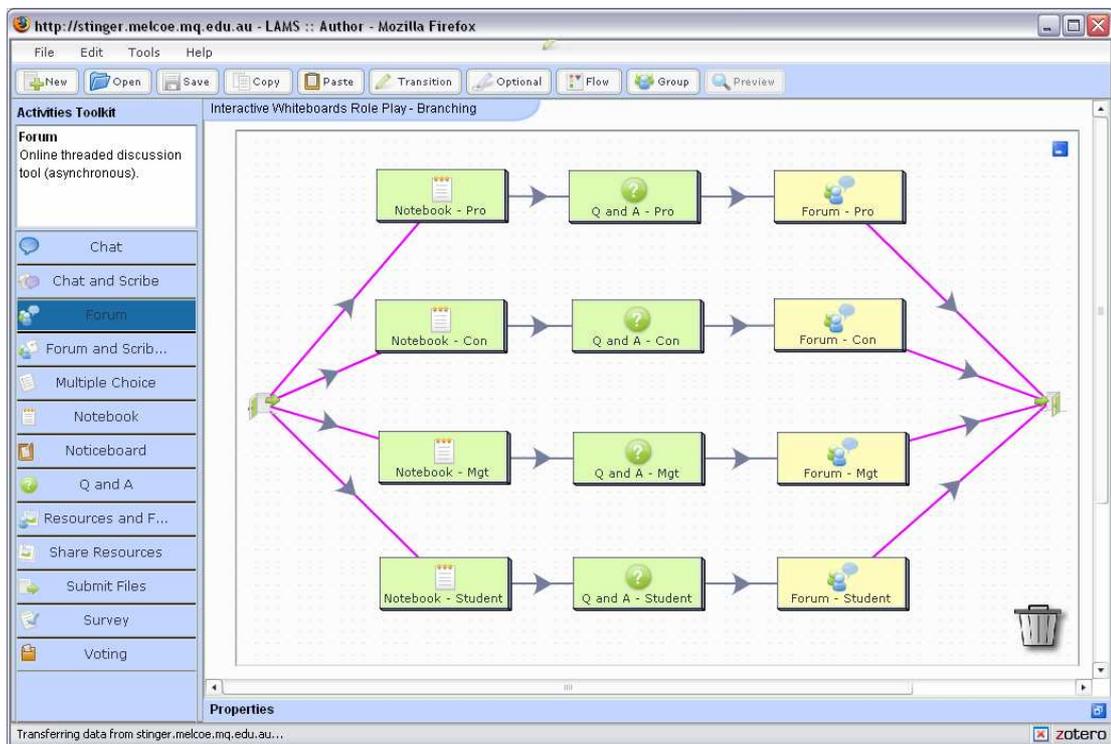


Figure 4. Screenshot of role play sequence, showing branched activities for each role group within Branching area (see Figure 2).

A negative effect was to limit the range of pedagogical approaches that could be represented using LAMS branching. Approaches such as having brief periods of small group work within an overall class set of activities was well suited to the LAMS branching approach, as the branched activities represented a discrete “chunk” of activities within the overall sequences; however, other approaches which involved students navigating many different pathways, potentially with different end points that did not “come back together” in a whole class activity, were not easily captured by this approach – for an example, see slide 14 of (Derntl, Parrish & Botturi, 2008 – this type of structure is sometimes referred to as the “Choose your own adventure” approach, in reference to a children’s book that allowed for many different possible endings to a story according to the reader’s choices during the story).

Hence, the positive effect was to provide a practical way of dealing with small-group branched activities within a larger class sequence structure, but at the expense of more sophisticated branching structures, especially those used for navigation of multiple pathways by individual learners (as opposed to groups).

While there have been other features added to LAMS V2 over time (such as new activity tools, video recording, naming of groups – for details, see <http://wiki.lamsfoundation.org/display/lams/Roadmap>) none of these features have a major impact on the nature of visualisation in LAMS authoring, and hence are not considered here.

5. Future developments

Two proposed future features are worth noting in the context of visualisation. The first is a possible redevelopment of the core Authoring environment using the Flex framework that will allow for a range of new visualisation features, including a “zoom in/out” feature at the sequence level, alternative activity tool palette representations (eg, clustered of activities according to the educational purpose of the activity tool, rather than alphabetically) and an annotation feature that will allow annotations to be added to any part of a sequence to provide additional educational explanations viewable at the sequence level (for an example, see slide 35 of Dalziel, 2008).

The most important impact of this redevelopment for visualisation issues is a new approach to branching where the branched activities are viewable (and editable) at

the same level as the main sequence – this avoids the current two-step process of editing branching inside a pop-up box that is separate from the main sequence view (and which hides the main view when open). This proposed new feature could allow for a more coherent perspective on the whole of a sequence of activities where branched activities play an important part in the overall design, as the detail of the branched activities will be viewable at the same time as the rest of the design (not in a separate box).

The second proposed new feature is the potential for completely different visualisations to the current “drag and drop” metaphor, but still mapped to the main sequence view. A range of possible new visualisations for LAMS were canvassed in a 2008 presentation (Dalziel, 2008), and in particular, a new “time-based” view of activities was proposed that provides a significant departure from the current visualisation approach (see Figure 5). This view is closer to the traditional Gantt chart of project management, but applied to educational activities.

There are challenges with the representation of time across multiple parallel educational activities, and this approach may not be very useful for asynchronous learning contexts – nonetheless, it is likely to have advantages for more synchronous-style activity sequences (such as those conducted in a computer lab with a class over a set time period as is common in K-12 school environments). While initially the representation would be a “read only” representation of the information in the main sequence view (with particular reliance on timing using “Stop Points” to determine the nature of the visualisation), it is possible that in time, an author could edit the time-based visualisation in certain ways, such as extending or shortening the time period of an activity by dragging the edge of the bar associated with the activity, with the resultant changes automatically reflected back into the timing within “Stop Points” in the main sequence view.

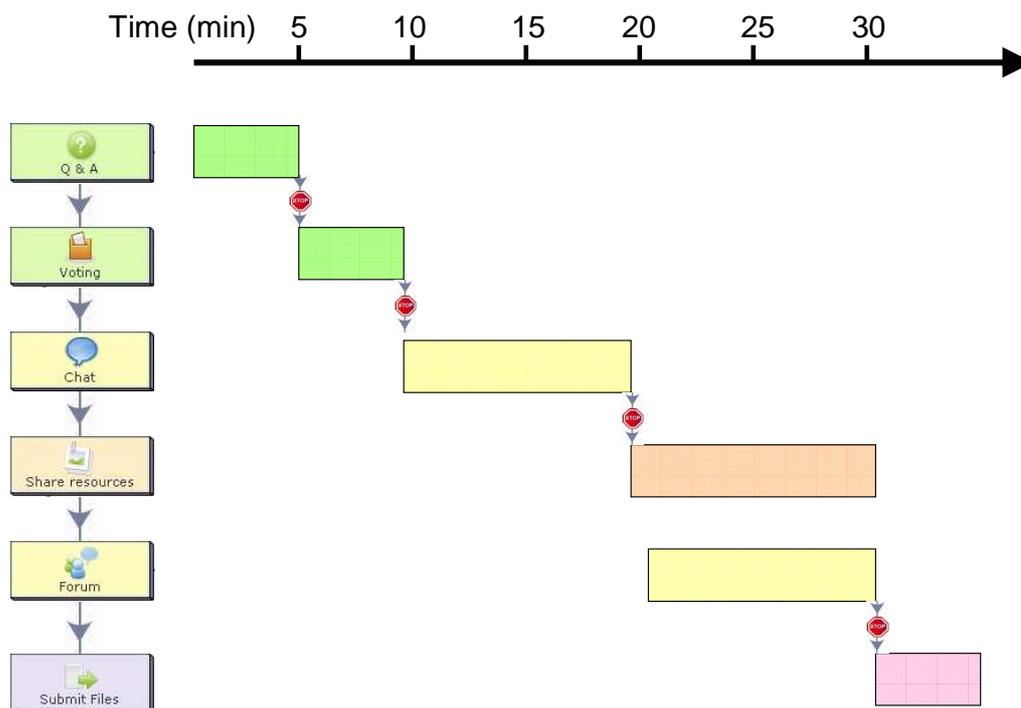


Figure 5. Proposed time-based visualisation for future LAMS authoring environment.

6. Barriers to adoption and visualisation

While LAMS has been of interest to a range of educators, this has not always translated into sustainable adoption. In a number of contexts, initial enthusiasm for LAMS among some educators has not then spread to a wider group, resulting in LAMS being seen as a niche rather than general e-learning technology (eg, use of LAMS in teacher training within universities, but not across all faculties). In other contexts, a successful trial of LAMS has not led to sustained use of the technology over the long term (eg, Masterman & Lee, 2005). There are several possible reasons for this lack of broader uptake of LAMS, such as challenges in “crossing the chasm” from adoption by innovators to the mainstream, as well as a tendency of some education IT support groups to focus on just the institutional LMS/VLE to the exclusion of other tools such as LAMS.

However, another possible reason for barriers to adoption relates to visualisation issues. Recent Learning Design work has identified the importance of providing pedagogical advice as an embedded component of Learning Designs – for example the use of the Conversational Framework in the LDSE project and the use of annotations in

CompendiumLD (Conole, 2009) to explain the purpose of certain activities, rather than simply provide their tool names (as in LAMS). While experienced LAMS users may be able to infer the pedagogical assumptions behind a LAMS sequence, most users only see the overt visualisation of tools and arrowed lines, and this does not convey the underlying pedagogical rationale of the sequence. It may be important in the future for LAMS to add additional layers of information (such as annotations) to provide a richer visualisation that represents not just the “how” of a sequence, but also the “why”.

7. Conclusion

The development of the LAMS Authoring environment, and its implicit visual language for Learning Design, is a product of key choices and constraints during to the development of the LAMS software, particularly the decision to collapse much of the flexibility implied in EML/IMS LD into discrete activity tools within a linear sequencing framework. Despite the limitations that arise from these decisions, a range of pedagogical approaches can be represented and run within LAMS and shared with other educators.

Notes

1. The name “Learning Activity Management System” was originally chosen as an explicit departure from prevailing Learning Management Systems and their features, in that LAMS had a different focus on activities and activity sequencing rather than courses and content. In practice, many educators found the name confusingly similar to Learning Management System, and often assumed the functionality was equivalent.

2. Please cite as: Dalziel, J. (2011). Visualising Learning Design in LAMS: A historical view. In J. Dalziel, C. Alexander, J. Krajka & R. Kiely (Eds.), Special Edition on LAMS and Learning Design. *Teaching English with Technology*, 11(1), 19-34.

References

- Campbell, L. M. (2002). Third meeting of the Valkenburg Group. Presentation for CETIS EC SIG, October 2002. <http://metadata.cetis.ac.uk/groups/20010809144711/FR20021105165633> (viewed on 1 Jun 2010).
- Conole, G. (2009). Learning design at the OU – visualising, guiding and sharing designs. Presentation for the 2009 European LAMS and Learning Design Conference, Milton Keynes, UK, 7th July 2009. <http://lams2009.lamsfoundation.org/abstract.htm#ou> (viewed on 1 Jun 2010).

- Dalziel, J. (2003a). Implementing Learning Design: The Learning Activity Management System (LAMS). In G.Crisp, D.Thiele, I.Scholten, S.Barker and J.Baron (eds.), *Interact, Integrate, Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education* (pp. 593-596). <http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/593.pdf> (viewed on 1 Jun 2010).
- Dalziel, J. (2003b) Introduction LAMS: An EML/LD "inspired" system. Presentation for the Fourth Valkenburg Group meeting, 23rd February 2003, Vancouver, Canada. <http://www.lamsinternational.com/CD/html/resources/presentations/Valkenburg.Feb03.LAMS.ppt> (viewed 1 Jun 2010).
- Dalziel, J. (2003c). LAMS – The Learning Activity Management System. Presentation for the CETIS Pedagogy Forum, 3rd August 2003. <http://www.lamsinternational.com/CD/html/resources/presentations/LAMS.Pedagogy.ppt> (viewed on 1 Jun 2010).
- Dalziel, J. (2005). LAMS in the future. Presentation for LAMS JISC Evaluation Conference, 11th January, 2005, Birmingham, UK. <http://www.lamsinternational.com/CD/html/resources/presentations/LAMS.JISCeval.AstonUniConference.Jan05.ppt> (viewed on 1 Jun 2010)
- Dalziel, J. (2006). Lesson from LAMS for IMS Learning Design. *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)* (pp. 1101-1102). <http://portal.acm.org/citation.cfm?id=1156068.1156419> (viewed on 1 Jun 2010).
- Dalziel, J. (2008). New Approaches to Visualising Learning Designs. Presentation for 3rd International LAMS and Learning Design Conference, Sydney, December 5th, 2008. <http://lams2008sydney.lamsfoundation.org/Presentations/Keynotes/Dalziel+Keynote+3rd+LAMS+conf.pdf> (viewed on 1 Jun 2010).
- Dalziel, J. (2010). LAMS Community Newsletter 85, 21st April 2010. http://www.lamscommunity.org/dotlrn/clubs/educationalcommunity/forums/message-view?message_id=1029904 (viewed 1 Jun 2010).
- Derntl, M., Parrish, P. & Botturi, L. (2008). Beauty and precision in Instructional Design. Presentation for ED-MEDIA 2008, July 2nd, Vienna, Austria. <http://www.slideshare.net/mikederntl/edmedia-2008-beauty-and-precision-in-instructional-design-michael-derntl> (viewed on 1 Jun 2010).
- Gibbs, D. & Dalziel, J. (2002) Creating Next Generation Interoperable Learning Tools. Presentation for NSW DET eCademy conference, 2002. <http://www.lamsinternational.com/CD/html/resources/presentations/MQ.e-cademy.ppt> (viewed on 1 Jun 2010).
- IMS (2003). Learning Design Specification. <http://www.imsglobal.org/learningdesign/> (viewed on 1 Jun 2010).
- Koper, R. (2001). Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML. <http://dspace.ou.nl/handle/1820/36> (viewed on 1 Jun 2010).

- Masterman, L. & Lee, S. D. (2005). Evaluation of the Practitioner trial of LAMS: Final Report. http://www.jisc.ac.uk/uploaded_documents/LAMS%20Final%20Report.pdf (viewed on Jun 2010).
- Olivier, B. (2004). Learning Design Update. http://www.jisc.ac.uk/uploaded_documents/Learning_Design_State_of_Play.pdf (viewed 1 Jun 2010)
- Takayama, Y., Ghiglione, E., Wilson, S. J. & Dalziel, J. (2007). Collaborative Activity Human Workflow (CAHW) in eResearch. Presentation for the NSF/Mellon Workshop on Scientific and Scholarly Workflow. Oct 3-5, 2007, Baltimore, MD. <https://spaces.internet2.edu/download/attachments/4788/Collaborative+Activity+Human+Workflow+in+eResearch.Final.pdf> (viewed on 1 Jun 2010)